IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

James Duncan Work

Application No.: 09/852,336

Filed: May 8, 2001

Examiner: Chea, Philip J.

Art Group: 2153

For: METHOD AND APPARATUS FOR INTERNET-BASED HUMAN NETWORK BROKERING

Confirmation No.: 4814

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail with sufficient postage in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

on August 3, 2005
Date of Deposit

Sarah M A Montgomery
Name of Person Mailing Correspondence

Signature 8/3/05
Date

DECLARATION OF JAMES DUNCAN WORK

JAMES DUNCAN WORK DECLARES:

1. I am the inventor of the patent application identified above and of the subject matter claimed therein. I have personal knowledge of the facts set forth herein, except where such information is stated to be based on information and belief and, in such cases, I believe such facts to be true.

2. I understand that, with the exception of claim 162, all of the claims presented in the subjectpatent application have been rejected under 35 USC 102(e) as being anticipated by Takacs, WO 01/077793 and that claim 162 was rejected under 35 USC 103 as being obvious in view of Takacs when considered in combination with "features well known in the art". I further understand that the effective date of the Takacs reference is April 7, 2000, that being the earliest priority date claimed in the Takacs reference.

3. I invented the subject matter of the present claims of this patent application prior to the effective date of the Takacs reference by (1) conceiving prior to the effective date of the Takacs reference said claimed subject matter; and (2) thereafter reducing to practice said invention prior to the effective date of the Takacs reference. Each of these activities was performed in the United States. Moreover, following such actual reduction to practice and with the assistance of my attorneys, I caused to be prepared and filed U.S. provisional application 60/ 203,374, from which the present application claims priority and which likewise supports the subject matter of the present claims.

4. In support of the above contentions I have attached hereto as Exhibit A a redacted version of a "Functional Specification" for Net Deva, a computer-implemented system for, among other things, reporting matches to searches initiated by a searcher so long as access control criteria are met, the matches including potential targets satisfying one or more search criteria defined for the searches, and the access control criteria (i) being selectably controllable by any of one or more persons in one or more chains of person-to-person relationships connecting the searcher and the potential targets, and (ii) defining attributes of such one or more persons and such persons' contacts that may be shared with others. I wrote this Functional Specification prior to the effective date of the Takacs reference.

4.1. At pages 13 – 14 of the Functional Specification I describe the basic functionality of providing matches to searches in accordance with access control criteria.

4.2 At pages 13 – 14 of the functional Specification I describe different types of access control criteria, including those based on connection strengths.

4.3 At pages 16 – 17 of the Functional Specification I continue the description of access control criteria, including those based on connection thresholds.

4.4 At pages 12 – 13 of the Functional Specification I describe access control criteria such as the need for personal communication before a search request will be forwarded or acted upon.

4.5. At pages 18 – 21 of the Functional Specification I describe the reporting of matches only so long as they satisfy access control criteria through a chain of users/relationships, the autonomous brokering of connections in support of the reporting of such matches, and

the use of trusted connection paths as means for satisfying access control criteria and user instructions related thereto.

5. Following my development of the Functions Specification, the subject matter described therein was implemented in source code and a run-time set of computer-readable instructions (object code) produced therefrom. Prior to the effective date of the Takacs reference, the object code was installed and executed on a computer system in the United States and under my control to practice the searches, the autonomous brokering and the reporting of matches to search criteria described in the Functional Specification and claimed in the present patent application.

6. Following the development of the computer software that implemented the Functional Specification I contacted my attorneys and together we developed the specification of my U.S. provisional application 60/ 203,374. That provisional application includes all of the same subject matter discussed above and a copy is attached hereto as Exhibit B for ready reference.

6.1. In particular, by April 28, 2000 I had provided my attorneys with materials describing my invention. See Exhibit C attached hereto, which is a copy of an e-mail communication in which I transmitted a copy of the Functional Specification and other materials to my attorneys.

6.2. By May 9, 2000, a draft of the provisional application had been prepared. See Exhibit D attached hereto, which is a copy of an email from my attorneys to me enclosing a draft of the provisional application for my review.

6.3 I subsequently reviewed and approved the provisional application for filing, which filing took place May 9, 2000.

I understand that willful false statements and the like are punishable by fine or imprisonment, or both (18 U.S.C. 1001) and that such willful false statements may jeopardize the validity of the subject patent application and/or any patent issued thereon.

Dated this 3rd Day of August, 2005 at Mountain View, CA

James Duncan Work

EXHIBIT A
To the declaration of James Duncan Work

Requirements and

Functional Specification

For Net Déva™

[redacted]

EDIT HISTORY:		
Draft 1.0	[redacted]	
Draft 1.0.1		
Draft 1.0.2		Added functional descriptions of the agents
Draft 2.0		Added object hierarchy, use cases, appendices

Table of Contents

IV. Overall Product Description

Net Déva is people-networking software that can be characterized as both a browser and broker of human networks on the Web: The Net Déva client is the browser; the Net Déva server is the broker.

Human networks are central to all value-creating activities and operate at multiple levels. They include:

- Personal networks – the personal and professional contacts each of us has.
- Organizational networks – links within and between organizations.
- Associations and interest groups – people attracted by common values, interests, and goals.

Net Déva users are existing online community members, members of organizational networks (independent consultants, alliances, partnerships, consortiums, associations) or employees of small to large companies. They engage in human development, organizational learning, training, participatory management, brokering, marketing, sales, trade, research, and consulting. They understand the value of networks and want new Internet tools with sustainable value that can give them an edge to make better human network connections on the Web.

With Net Déva users will be able to:

- Quickly narrow the choices in finding and evaluating new partners, clients, colleagues, suppliers, employers, employees and information sources by learning enough about them to properly evaluate.
- Get recommendations and introductions from a trusted source to build extended networks based on trust and value.
- Screen incoming information and requests and maintain privacy when desired.
- Connect with more of the right people and clients to foster new relationships

Net Déva offers users a rich environment to create and maintain all types of human networks supported by online interactions. The ***Net Déva client*** is a Java applet that works within Microsoft Explorer and Netscape Navigator web browers offering a rich profiling environment for sharing information. The ***Net Déva Web application server*** is comprised of gatekeeper and network broker intelligent agents that are trainable to emulate the functions of a human broker, making highly accurate matches while protecting personal privacy.

Net Déva works like this: First users create rich profiles needed for greater visibility and more accurate matching and evaluation of new connections. These profiles are both locally stored and uploaded on Net Déva application servers that will reside on an intranet or a Web server for an online community. Users instruct a personal gatekeeper agent to control access and a personal search agent to find desired connections. A network broker operating on the server is then used to evaluate matches and broker relationships. All parties can trust the broker to follow their instructions regarding the match desired and the degree of privacy desired.

Net Déva offers a better way of matching people's interests and building personal connections to the right people and information they value. Getting the right person's ***attention*** is what really matters and is a necessity for creating almost any type of value. However, in the real world, the factors that influence individual and group attention are very complex. Net Déva's sophisticated agents will build on the complexity of human relationships in ways that are transparent to the user.

Net Déva also addresses the flip-side of attention by ***protecting personal privacy***. Privacy is an enormous issue on the Internet and will continue to grow in importance. Net Déva addresses privacy by managing access to attention because we understand that the more visible it's possible to be, the more it becomes necessary to be ***selectively visible.***

It may seem remarkable to include all these capabilities in one product, however, the reality is that for a product to have ***real sustainable value*** it will have to offer these key components. However, our software development plan involves using Net Déva agents in ways that can be plugged-into other existing applications, in addition to their use as part of the entire Net Déva system.

The Net Déva components are:

1. **Profile Builder**
   A tool that helps people and organizations reveal their capabilities, projects, goals and values so that others can adequately evaluate them. Net Déva's profile builder is rich and customizable, asking people what they want to accomplish, and then helping them build a profile that can accomplish those goals. Each individual has total control over what goes into his or her profile, who has access to it, and what networks to connect it to. This is significant because in order to make good connections it's as important for others to evaluate us as for us to evaluate them.

2. **Personal Gatekeeper Agent**
   Allows people to protect the information in their profiles and their attention from inappropriate access, and make these personal profiles connectable. Is easy to use and customizable allowing access categories that can be

assigned at either macro or micro levels within one's profile. The Gatekeeper Agent will ultimately be trainable – learning from both direct instruction by its owner and by "observing" it's owner's behavior.

3. **Personal Search Agent**
Guides the user in constructing a profile for a search target.

4. **Network Broker Agent**
A network agent that emulates the function of a human broker negotiating between users personal search and gatekeeper agents.

4. **Verification Agent and Network**
A network agent that authenticates and verifies information people have recorded in their profiles and designed to work in conjunction with systems like truste www.truste.com. Using Net Déva, we and our partners will develop a Verification Network to verify Net Déva profiles. This will *simplify and strengthen* the verification process for people offering their services (e.g., job seekers or consultants), for people seeking services (e.g., employers or prospective clients), and also for people who verify others (references given by job seekers or consultants).

See Appendix A for a scenario illustrating how Net Déva is used.

V. Architecture Overview

Net Déva is a multi-tier system consisting of

- A Java applet client which can be downloaded from the server and resides on the user's local system. The client contains a local database as well as client-side agents of the following types:
    - Profile Builder agent
    - Gatekeeper agent
    - Search target profiling agent
- an HTTP server,
- a Java application server (which can be combined with the HTTP server), which includes the following Net Déva's agent applications:
    - Search agents
    - Gatekeeper agents
    - Network Broker agents
    - Verification agents

- a database server, containing a secure, composite data structure which maintains information about all users of the system plus a record of prior searches and matches which the Network Broker agent can access to learn and reapply successful search strategies.

VI. Platform Requirements

The bulk of the system will be built as Java applets to maintain the richness of an object-oriented approach while using a standard web browser and HTML as the delivery platform for the user interface. For compatibility with Java-based interface agents that might be delivered through the user interface, a Java application server should dynamically generate the HTML ("compiled HTML"). [redacted]

We prefer that database storage is also object-oriented rather than relational, unless we find that cost and/or performance are prohibitive. There must be a high capacity database on the server, and more limited "persistent store" capabilities on the client. [redacted]

VII. Net Déva Objects

A. NetDevaClient

General Description

Software for the client is distributed as a package containing:

- A Java applet or applets
- Persistent store for the applets
- A web browser (installation optional if the user already has a browser which supports the current version of the Java virtual machine).

The applets and local database will installed in the Registry of the operating system, and stored in a named directory. [redacted]

Required Functionality

Locally stored applets conduct a dialog with the user to build the profiles off-line. Data resulting from this dialog is stored in a local database. **This database must be encrypted** or otherwise secured against prying eyes

and file theft, since client machines in any organization are much more loosely guarded (if at all) than enterprise-level servers. A "local gatekeeper" ensures that only data that the user has designated to be shared gets sent to the permanent storage on the server. Search queries may also prepared off-line, before being sent to interact with the server's broker agent.

Object Oriented Database

We feel that an object oriented database will be important for both the client and server due to it's superior integration with Net Déva's object design and due to the extreme complexity and flexibility of matches and links between objects that will be required. [redacted]

Interface Personalization

It will be extremely important for the interface to respond to differences in user objectives, style and context, and to changes in these over time. To accomplish this, users will be queried on their objectives and preferences in their initial session with Net Déva (in an "Orientation Interview" conducted by the user interface) and in subsequent sessions. (See the document "Screens – Orientation Interview.doc" located in the attached file referred to in Appendix D, for a draft of this section.) Advanced functionality will also include monitoring user's behavior to detect style and preferences. Information about user objectives and preferences will also be stored in the **UserStatus** local database. Also stored there will be the user's profile completion status, information about how the user has used Net Déva and results of use, and possibly also user satisfaction with results.

Information on the user's objectives and context (e.g., industry or profession, country, etc.) will be used to select and customize prompts to present to the user, and (in advanced versions) to make suggestions or offer information to the user. For example, information collected on the user's organization type and profession will be used to select prompt variants. Information on the user's objectives will be used to guide the user to complete sections of the profile that will be most necessary for achieving those objectives, and to optionally skip sections that are not essential for his objectives. Information on the user's completion status will also be used, along with user objectives, to guide the user to complete

the next most important section when he logs in next. User completion status may also be used to reward users for profile completion and for the value that this provides to other users.

The interface will also respond to differences in people's style and preferences. For example, a task-focused person who wants to cut to the chase and solve an immediate problem will get a more minimal set of initial questions than a more curious or expressive person who wants to carefully construct his profile. In the case of the quick-start person, the interface will cut the orientation and profile building to a minimum and quickly find what the user wants to do. For example, if the user wants to find a particular type of contact. Instead of asking him a lot of questions about himself, a "search deva" (search profiling agent) will ask him to profile the person he wants to contact. This will get the user familiar with the basic elements of a profile. Then the deva will remind him that that person he's looking for will probably need to know some things about him. This will give the interface a task-focused reason to get the user to start profiling himself. Then in later sessions the interface can gradually get the quick-start user to fill out more of his profile. For example, when results come back from a search and a good prospect wants to know more about him, then the user will have to reveal more in order to complete that connection.

This type of personalization will require an intelligent, dynamic and non-linear interface. Net Déva object designers will work with Equinox team members to develop object code that contains the intelligence (rules, etc.) needed to respond to user preferences, context, and objectives. Reference to some of these rules can be seen in the screen design documents in referenced in Appendix D. Since each question presented to the user in each profile section will be an object, the interface will need to dynamically select which questions and prompts to display, how to number them, etc., based on accumulated information stored in the UserStatus database.

Client Components

See Appendices B – E for illustrations, conceptual demos and draft design notes related to the interface. [redacted]

A basic object hierarchy for the Net Déva client is shown below.

1. InterfaceToUser

This object will contain the intelligence described above under "Interface Personalization" and also the interface objects needed to feed the local database. See also Appendix C – Client Menus.

2. LocalDB

Local database. Sections of the database include:

a) UserStatus

Described above under "Interface Personalization."

b) OrgProfile

The organization profile will be completed once for each organization and will be accessible to each user's Net Déva client in that organization. This profile may be completed by a single person or contributed to by multiple people. Some information included in the profile will be dynamically added by Net Déva based on collective responses of organization members. Sections of the database include:

(1) Capabilities

(2) History

(3) Values

(a) Culture

(b) BasicValues

(4) Goals

(5) (Projects)
- collected from member profiles, can also be added by system administrator)

(6) (Networks)

– collected from member profiles, can also be added by system administrator

(a) ProfileOfNetworks

(b) Contacts

(i) Internal

(ii) External

(c) Resources)

c) PersonalProfile

The sections of the PersonalProfile are very similar to the OrgProfile. Both objects will inherit from an abstract Profile class. The sections of the profile are described below. See the attached demo NDSMTLK.DBD referenced in Appendix C for an illustration of these sections. See also the documents Screens - Pers Profile 1.doc and Screens - PersProf Prof Net & Resources.doc in Attachment D.

(1) Capabilities

(2) History

(3) Projects

(4) Values

(a) Interests

(b) Style

(c) BasicValues

(5) Goals

(6) Networks

(a) ProfileOfNetworks

(b) Contacts

(c) Resources

(7) ContactProfiles

This database will store profiles of the user's contacts that were either entered by the user or downloaded from a Net Déva server. If the profile was downloaded from the Net Déva server, the profile will contain a link to the copy of the profile stored on the server so that the profile can be updated when it is accessed (as allowed by the profile owner's Gatekeeper Agent).

(8) Gatekeeper

This database stores general and specialized Gatekeeper instructions and a log of Gatekeeper actions and user responses to these actions (e.g., satisfaction, correction, etc.). These will be used to train the Gatekeeper Agent. Note: The access and security codes used by the Gatekeeper Agent will be stored in the user's Profile.

(9) Searches

This database will store prior search parameters, results, and user responses to results so that searches may be reused, modified, and improved.

3. InterfaceToServer

This component will allow the user to upload profile sections and agent instructions to the server and download results and other communications from the server. A database replication and file optimization scheme will also be included. [redacted]

The interface to the server will be closely connected to the client-side gatekeeper agent.

4. ClientGatekeeper

The client gatekeeper will insure that data marked with the access code, "Self Only" will not be shared with the server. The client gatekeeper will also respond to requests by the server for information stored only in the local database, or for specialized responses to search results, e.g., requests for additonal information or actions by the user. Advanced functionality will include the

ability to filter all types of incoming information and requests for the user's attention, including email.

B. Servers

1. NetDevaServer

General Description

The Net Déva Server will be a Java application server that will include the functions of an HTTP server, dynamically serving HTML content and associated Java applets to the client. It will also host Java applications which serve the functions of Net Déva's server-side agents, including the Network Broker agent, Gatekeeper Agents, and Verification Agent. It will interface in a secure fashion with the database server.

Required Functionality

Net Déva Client Applets will be stored on the Java application server, and delivered to the client on demand in the context of HTML pages. Since these applets may also be stored on the client, the server queries the client for to find out if the most current version of the applet is present on the client, and if not it provides it.

A Java application communicates with the data structures when the client gives it new information to store there, and updates the server-side gatekeeper for that user with new instructions for maintaining secure access.

The Java server also accepts search agent instructions, and communicates these to the broker agent. Upon a successful match to the criteria of the search, the server communicates back to the client regarding the successful path to the repository of information or contacts that satisfy the search.

Desirable Advanced Functionality

It is desirable that the server be able to request new information (not currently stored on the server) from the clients to see if a match is possible based on information not yet shared; this could conceivably lead to human intervention and negotiation toward selective release of the information. [redacted]

Data Structures

Applets will be stored in directories on the server, individually and as packages to be downloaded to new users.

Most of the server-side data structures will be stored on a database server.

Server Components

a) HTTPServer

b) InterfaceToDBServer

c) InterfaceToExternalServers

d) NetDevaAgents

(See Use Cases in Section VI for more detail.)

(1) NetworkBrokerAgent

The Network Broker Agent has to search the User Profile database (**UserDB**) to look for matches against the criteria specified in search parameters sent by a Net Déva client. It then has to evaluate the closeness of fit to the search parameters. If the search parameters specify connection criteria, such as level of trust, type of connection, etc., then the Broker Agent may have to discover and evaluate connection paths between the searcher and the prospective target. For each prospective target found, the Broker Agent next has to ask and receive permission from the target's personal server-side Gatekeeper Agent for release of requested information, which is then sent back to the requesting Net Déva client.

(2) PersonalGatekeeperAgent

The server-side Personal Gatekeeper Agent will evaluate any request delivered by the Network Broker Agent and determine what information may be released. This will be the case in response to both searches and browsing functions initiated by Net Déva Clients. The server-side Gatekeeper Agent may also request any additional

information it needs in order to better evaluate what access level to give to the request.

(3) NetworkVerificationAgent

The Network Verification Agent responds to any updated verifier information sent by a Net Déva Client. The Verification Agent will send an email message to each verifier listed asking the verifier to confirm the Client supplied information to be verified. The Verification Agent will also receive replies to these emails and evaluate them. Finally, the Verification Agent will place a "verification stamp" in a section of the requesting Net Déva's Client's profile containing the results of the evaluation. This verification stamp will be editable only by the Verification Agent. It may not be edited by the user or any other entity or application.

(4) Host for InFlow™ and Other Applications

InFlow™ is an application for social network analysis that will be licensed from a Net Déva strategic partner. Other third-party applications may also be included.

e) FileServer

(1) ClientApplet

(2) Other files

2. DBServer

a) UserDB

This database will store all the profiles and instructions uploaded by Net Déva users. Database sections include:

(1) UserProfiles

(2) GatekeeperInstructions

(3) SearchInstructions

b) SearchResults

This database will store results of searches to be used by the Network Broker Agent in refining and reapplying it's successful search strategies. Personal Search Profiling Agents operating on Net Déva clients may also access this database in order to recommend search strategies to the users and prompt for information needed by these strategies.

c) ExternalServerIndex

This will be an advance version database that will help a server extend searches to external Net Déva Servers. [redacted]

VIII. Use Cases

Below is a list of use cases [redacted]

A. User Builds Profile

See attached demos and screen design notes for illustration.

B. User Instructs Gatekeeper

See attached gatekeeper interface screen design notes for illustration and detail. [redacted]

C. User Uploads Profile and Gatekeeper Instructions

D. User Creates and Launches a Search

(This action is illustrated in NDSMTLK.DBD (DemoShield demo) referenced in Appendix D.)

User presses Search icon or menu item.

Interface displays Search parameters screen.
This screen is divided into the following sections (see illustration in demo):

- Nature (bus/prof, personal)
- Objectives (Action/Object; Description)
- Compensation Requirements

- Target Profiles
  - Organization
  - Individual Contact
- Connector Profile
- Verification/Search Path
  - Connection Types and Strengths
  - Max. connection levels
  - Reference check instructions
  - Net Deva Certification type
  - Other verifiers

User can enter an English description of the Search Nature. This can be used for a human readable description of the search – by prospective targets, connectors, or human brokers helping with the process. Advance functionality may also include machine parsing and comprehension.

User enters a pre-selected category describing the search by specifying an Action/Object pair, e.g., "Find Partners" or "Offer Services", or "Exchange Ideas". These categories will be used to help optimize search results and also to group searches for refining search strategies, and to retrieve stored searches.

User enters compensation requirements, if any exist. This includes compensation the user is willing to pay to targets or connectors or compensation that the user expects if offering services or goods.

User profiles the target organization and target person to specify the skills, capabilities, and values that sought. The interface for doing this will be similar to organization and personal profiles, except that there will be a place to indicate the relative importance of any profile criteria specified – e.g., required, very important, important.

User profiles likely connectors to the target. This is optional but will be extremely important if the type of connection to the target is important, for

example, if the user wants a trusted recommendation and introduction to the target.

Advanced functionality will include a Search Profiling Agent that will help the user specify the kinds of profile information that are most likely to result in matches for the desired search objective. For example, the Profiling Agent may be able to recommend what type of connectors to look for, or what kinds of information the target is likely to need in order to respond.

User enters Verification parameters, if any exist.
The Verification instructions will help the broker agent plan its search path. I.e., does the user want to limit the search to:

Your own closest connections?
The closest connections of your closest connections?
The people in your organization?
People in allied organizations?
Connections of people in allied organizations?
The global universe accessible by Net Deva?
Only Net Déva verified or certified targets?
etc.

If the user is in a hurry and leaves out important sections, the Search Profiling Agent will give the user reasons for completing those sections and ask them to do so.

A search may also be initiated or saved for later use whenever the user adds or modifies his or her own profile information regarding projects, goals, or interests. Projects especially relate to searches if the project has requirements that aren't yet filled. After entering project information, user may be prompted, "Do you want to start a search for project requirements?" (Or else can push the "search" button at the bottom of the project requirement screen.) If yes, the project specifications will be used by the agent as the basis of a search and user will be prompted for additional instructions needed to carry out the search. Likewise the user can initiate a search for people who share common interests, values, goals, or background.

User then presses the search launch icon to launch the search. The search is first launched on the local system to look for matches or likely connectors among the user's own locally listed contacts (including locally listed contacts of other members of the user's organization); next the search is uploaded to the Net Déva Server.

E. NetDevaServer Responds to Search

(This action is illustrated in NDSMTLK.DBD (DemoShield demo) referenced in Appendix D.)

The Net Déva Server receives search instructions sent by a Net Déva Client.

The Net Déva Server instantiates a Net Déva Broker Agent to carry out the search.

The Broker Agent includes these objects:

- Interface to Server
  - Receives search parameters
  - Responds with search results
- Search agent
  - Interface to User Profile and Gatekeeper databases
  - Interface to Search Results database
  - Search strategy developer and executor
  - Search result evaluator

The Net Déva Broker Agent parses the search into it's component parameters and conducts a search in the User Profile Database located on the server to try to find best matches.

If the search parameters include connection types and strengths, then the Broker Agent will seek connection paths that match connection parameters. This may include strategies such as:

> Starting with targets and working backward to try to connect to the searcher, via likely connectors.
>
> Starting with the searcher's contacts and working outward to try to connect to targets or other likely connectors.

Once matches to search parameters are found and evaluated, the Broker Agent will start with the best targets and likely connectors and negotiate with the target's Gatekeeper agent for release of information about the target to the searcher.

The target's Gatekeeper Agent will evaluate information which the searcher has allowed the Broker Agent to reveal in order to determine what level of access to assign to the searcher's request.

Based on the access code assigned to the searcher by the Gatekeeper, the Gatekeeper will give the Broker Agent permission to report back to the searcher any requested information that has a security code equal to or lower than the searcher's assigned access code. In some cases this will be all information requested, in other case it may include some information but not include specific contact information (name, etc.); in still other cases it may be no information.

If the Gatekeeper is interested in the request but cannot assign a high enough access code to the searcher to release the information requested, the Gatekeeper agent may (if previously instructed by it's owner) ask the Broker Agent to query the searcher for the additional information that it needs to release the requested information. This request for more information will then be relayed to the searcher's server-side Gatekeeper which will decide what to do with the request. For example, the searcher's server-side Gatekeeper may a) supply the information, b) deny the information, c) send the request back to the searcher's client to request action of the searcher's client-side Gatekeeper or directly of the searcher himself. Such additional information supplied by the searcher will then be relayed back to the target's Gatekeeper for re-evaluation of access.

If the target's Gatekeeper responds negatively (or sub-optimally) to a searcher's request for information, the Broker Agent will then attempt to find a trusted connection path between the searcher and the target (if it has not already tried to do this). If a trusted connection path is found, then the Broker Agent will submit this additional information to the target's Gatekeeper Agent to try to improve the access assigned.

When the Broker Agent is looking for likely connectors to targets, the Broker Agent will be asking the connector's Gatekeeper Agent for permission to search the connector's contacts for targets or other likely connectors. This will allow the Broker Agent to conduct extended searches through multiple "degrees" of connection.

Once results are obtained from target or connector Gatekeeper agents, the Broker Agent will collect all results obtained and rank and report them back to the searcher. The report back to the searcher will include:

<u>"Direct Hits"</u>

1. A ranked list of "direct hits" (people, orgs., info that matches the search target).

2. Hyperlinks to all relevant evaluation information that is accessible to the searcher.

Connectors

1. A ranked list of potential connectors to the target. The highest ranked connectors will usually be people who are most likely to have the strongest connections to "direct hits" and who also have strongest connections to the searcher.

2. Hyperlinks to all relevant evaluation information about connectors that is accessible to the searcher.

Messages and Requests

1. Messages from potential targets or connectors, such as, "Please contact me personally for more information--or for a personal introduction."

2. Requests from potential targets or connectors--asking for more information required to allow more complete access

Since searches can be quite complex and since User Profiles will contain varying degrees of information related to desired parameters, we will be doing research to develop search strategies especially suited to these conditions. We will research various strategies and refine and compare them. Potential strategies include:

a. 7G adaptive network (neural network) technology to evaluate and optimize complex, fuzzy matches, and which can make use of weighted connections within search paths.
b. Advanced 7G capabilities.
c. Modification and use of other third-party search engines. For example, we may parse and store user profile parameters in text files which can then be grouped according to parameter category and individually indexed and searched using available search tools. This will help improve the accuracy of matches by constraining searches to desired categories.
d. Development of custom built object-oriented search strategies and technology.

F. Client Gatekeeper Responds to External Communication

This use case is partially addressed in E, above, and elsewhere, but will be elaborated upon later. It will include Client Gatekeeper responses to server requests and advanced feature response to email.

G. User Lists Verifiers

This use case will be fairly simple and will entail a method to list verifiers for an entire profile as well as profile sections.

H. NetDevaServer Verifies User Profile

[redacted]

I. User Browses Contact Profiles

This use case will involve actions of the client browser, the network broker, and target Gatekeeper agents.

J. User Distributes Profile Info

This will be advanced functionality that will be similar to a search, but with the intent of finding receptive receivers of information published by the user rather than requests for information from targeted users.

[redacted]

X. Appendices and Attachments

Appendix A -- Scenario – Julia's Search

How Net Déva Works

Scenario

Julia's Search

Note: user interface screens are for demo purposes only

Introduction

Making the **right** new people connections involves three basic steps, each more difficult than the prior one.

1) Getting a **list** of candidates.
2) Selecting the most **qualified** candidates – the ones with capabilities and experience more finely tuned to your needs.
3) Getting a target's **attention**.

The first step can be done readily with a good directory or database; but that's about all they can do. Net Déva addresses the more difficult steps.

Julia's Problem

Julia Ingersen was a senior consultant in a mid-sized consulting firm in New York City. The company has offices in Chicago and LA, and close alliances with other companies in London, Brussels, and Tokyo.

Julia had targeted as a prospect an international company that she had never dealt with before. To make any progress, she knew she needed access in the form of a high level introduction,

and probably also some inside information. She had sent a message and material and made a follow-up call to the target, but the logical contact there was overwhelmed with work and other requests for his ***attention***. He took a superficial look at her materials, said *"Sorry,"* and then didn't want to talk anymore.

Six months ago, before getting Net Déva, she knew that her options would have been to either give up or spend a lot of time trying other avenues, possibly with no result. Her best bet would have been to go through her Rolodex and find people she knew who might give her a lead and then call them, explain her need each time, and then follow-up on any leads, which might generate other leads, as well as many blind alleys. Sending a broadcast email or posting a message on the company's discussion server sometimes helped, but too often it either returned little information, or leads that didn't really match her needs.

With Net Déva, she knew that she could explain her needs ***once*** (to her personal Net Déva Search Agent) and would then get the best leads available, without depending on getting her colleagues attention or full understanding of what she was looking for. If one of the 400 people in the company's alliance network had a good connection to her target, Net Déva would probably find it.

The Solution

Julia's network began using Net Déva six months ago, so Net Déva was not yet at its full capacity since people were still adding information and contacts. Fortunately, though, Net Déva doesn't require that all members of a network be equally thorough in entering information. In this case, after six months, 3/4 of members (300) had at least ***profiled*** themselves and their networks and entered a few specific contacts. Her network chose an option to automate much of the data entry involved in order to reduce people's time, so almost two thirds of the 300 participants had also entered a substantial number of contacts – about 50 each. Thus, there were about 10,000 contacts in the network, of which over 8,000 were unduplicated.

This is not a huge number of contacts, however, in six months there would be many more, plus the firm was providing incentives to its members who were able to get several of their external contacts to also use Net Déva. This incentive program was already having good results in increasing the size of the extended network.

The firm was also part of a larger industry association that was offering Net Déva to its members, and that was having even greater effects. As a result, the total number of contacts in the ***extended network*** could easily exceed 500,000 within another six months and would keep growing exponentially.

But, back to the present. . .

Eight thousand unduplicated, high value, high access contacts is substantial material for Net Déva to work with, especially since the network profiles contributed by participating members also provide considerable information needed to point to likely connectors to most targets.

Here are the steps Julia took to prepare and launch her search:

1. She entered a summary description of her search i.e. Who/what she wants to connect with, for what purpose

2. She profiled the target. In this case she had the target's name and also entered a profile including other important criteria: location, industry, specific type of company and products.

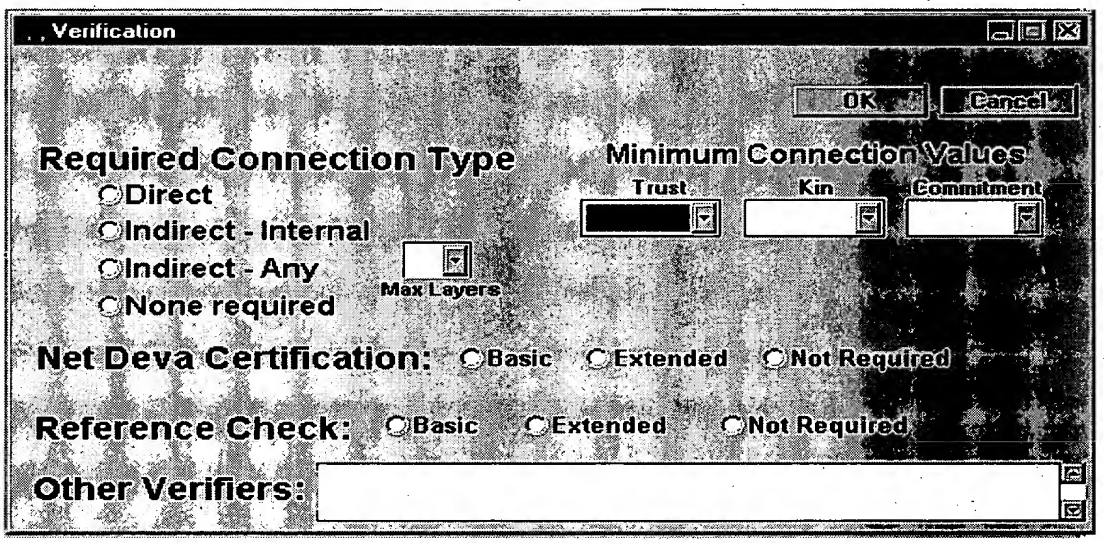
3. She profiled particular target contacts names of people, roles, and other factors like professional interests that would increase the likelihood of a good match.

4. She profiled likely connectors to the target for example, specific or general types of members of the target's network: clients, suppliers, partners, advisors, or other associates.

5. She indicated level and types of connection/verification desired. In this case she specified that she wanted only close, trusted contacts, and a maximum of two levels of connections.

6. She specified desired results: She wanted both connectors to this target as well as to other potential targets.

7. She launched the search (and also saved it for future use in similar situations).
Net Déva then implemented the search on the alliance network server.



Results

1. Actual connectors to this target:
Jane Adams, Chicago Office has a close connection to a V.P. in the target organization.

2. Likely connectors to this or similar targets.
Names of four people in the network who have close contacts in this industry and with this type of company, or with likely clients of this type of company.

3. Messages from connectors:
"Call Jane before you call her contact." Julia was quite happy with these results, especially since they didn't take that much of her time or anyone else's time. With these results she could now make just a few phone calls to exactly the right people.

Appendix B – Client Menus

Top Level Menu

- File
  - Open
  - Save
  - Save As
  - Print
  - Send
  - Get
  - Configure
  - Exit
- Profiles
  - Personal
    - Self
    - Others
  - Organizational
    - Mine
    - Clients
    - Allies
    - Others
- Networks
  - Personal Contacts
  - Org. Contacts
  - Resource Networks
  - Personal Network Profile
  - Org. Network Profile
- Agents
  - Gatekeeper
  - Search
- Applications
  - Explore
  - Resumé
  - Strengths
  - Connectors
  - Verify
  - Map & Analyze
- Help
  - Overview

Contents
About

Personal Profile Menu

File
- New
- Get
- Send
- Save
- Print
- Exit

Capabilities
- Domains
- Skills

History
- Experience
- Education
- Accomplishments
- Lessons learned
- Background
- Timeline
- Stories

Projects
- Professional
- Personal & Social

Goals
- Professional
- Personal & Social

Values
- Interests
- Style
- Principles & Beliefs

Networks
- In my Org.
- Clients
- Allies
- Other Professional
- Personal & Social

- Network Profile
- Resources
- Locations

Help
- Overview

Org Profile Menu

File
- New
- Get
- Send
- Save
- Print
- Exit

Capabilties
- Domains
- Products
- Services
- Principals & Staff

Background
- Accomplishments & Testimonials
- Size
- History
- Stories

Projects
- Internal
- Collaborative

Goals
- Long-term
- Near-term

Values
- Culture
- Principles

Networks
- Internal
  - People
  - Departments
  - Teams

Allies
Clients
Other
Network Profile
Resources
Locations

Help

Overview

Appendix C – Early demo of profile and search screens

See attached file, NDSMTLK.ZIP. This zipfile contains:

NDSMTLK.DBD -- excerpts from a DemoShield Demo done on Net Déva two years ago,

DEMO.EXE and DS.DLL – DemoShield viewer files.

To view the demo, place all three files in a single directory and the issue the command: DEMO.EXE NDSMTLK.DBD.

These demo excerpts illustrate Net Déva profile screens and search parameter screens.

Appendix D – Screen design notes

The Attached zipfile, NDCLIENT.ZIP contains the following Word documents that contain design notes for Net Déva Client Interface screens. They should be reviewed in roughly the order shown:

Screens – Orientation Interview.doc

Screens – Gatekeeper.doc

Screens – Org Profile.doc

Screens – Personal Profile 1.doc

Org Domains.doc
(Capability domain categories)

Functions and Roles.doc
(Capability function and roles categories)

Screens – PersProf Prof Net & Resources.doc
(Personal Profile Network Profile and Published Resources screens)

Notes – History.doc
(Rough notes for elements to include in Profile History and Background screens)

These documents are for illustration of design concepts [redacted]

A partial test implementation of some of these design notes as a Java applet can be viewed at http://www.fred.net/hawk/netdeva/

Appendix E [redacted]

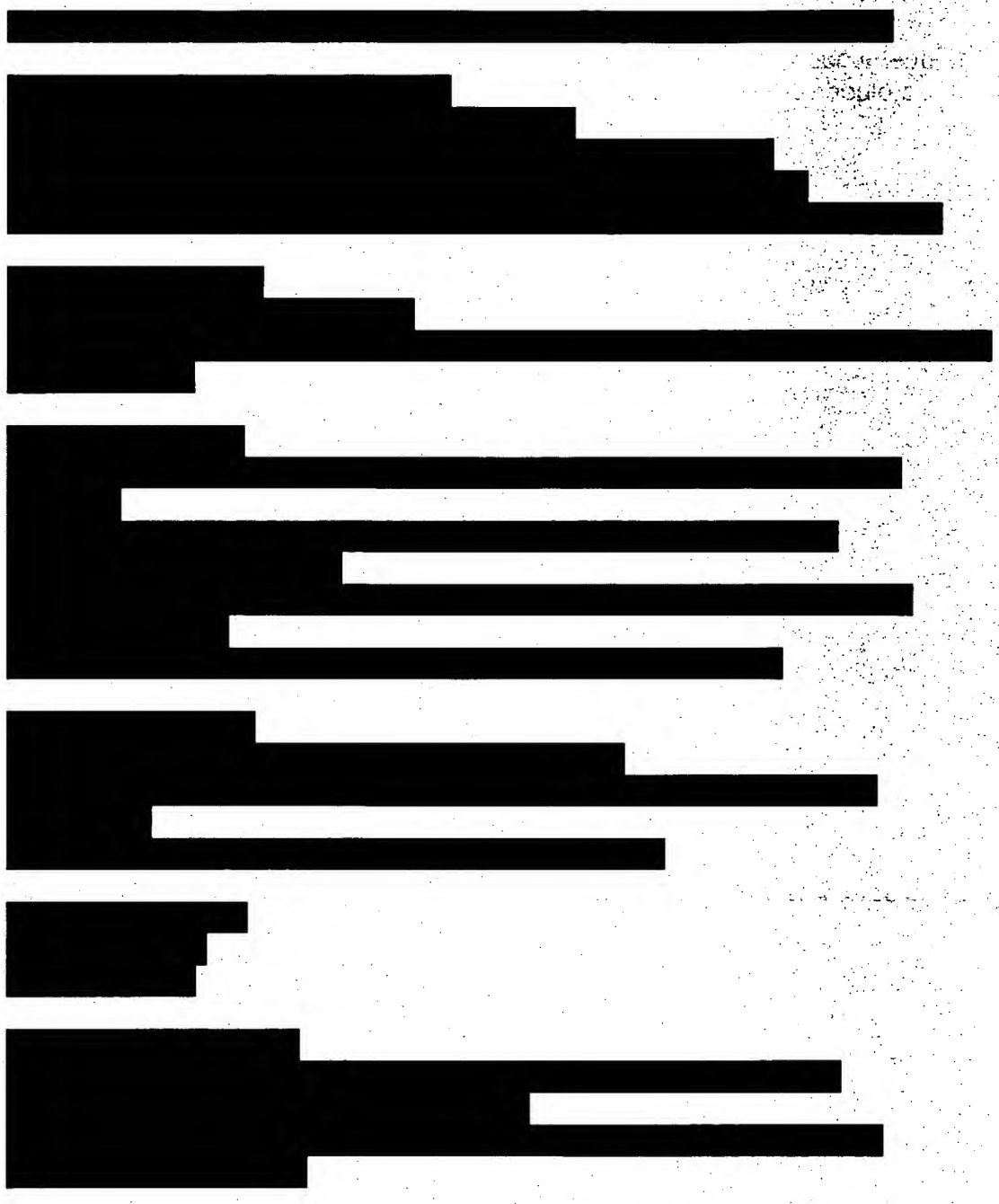


EXHIBIT B
To the declaration of James Duncan Work

Attorney's Docket No.: 004938.P001Z Patent

United States Provisional Patent Application

For

METHOD AND APPARATUS FOR INTERNET-BASED HUMAN NETWORK BROKERING

Inventors:

James Duncan Work

Prepared by:
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1026

(408) 720-8300

"Express Mail" mailing label number:_____
Date of Deposit:_____
I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Assistant Commissioner for Patents, Washington, D. C. 20231

_____
(Typed or printed name of person mailing paper or fee)

_____
(Signature of person mailing paper or fee)

_____
(Date signed)

METHOD AND APPARATUS FOR INTERNET-BASED HUMAN NETWORK BROKERING

FIELD OF THE INVENTION

The present invention relates to a "people-networking" scheme that may be embodied in computer software and/or hardware and that can be characterized as both a browser and broker of human networks on the World Wide Web, the graphical user interface portion of the Internet. When used in a computer network environment employing a client-server architecture, a client-side software application may act as a while a server component may acts as a broker.

BACKGROUND

Human networks are central to most, if not all, value-creating activities and operate at multiple levels, including:

- Personal networks – the personal and professional contacts each of us has.
- Organizational networks – links within and between organizations.
- Associations and interest groups – people attracted by common values, interests, and goals.

Today, many individuals may be regarded as existing online community members, members of organizational networks (independent consultants, alliances, partnerships, consortiums, associations) or employees of small to large companies. They engage in human development, organizational learning, training, participatory management, brokering, marketing, sales, trade, research, and consulting activities, all of which depend, to some degree, on inter-human networks. Such individuals generally understand the value of computer networks as tools for sharing information,

but presently these individuals have only limited access to tools that can give them an edge (e.g., a competitive advantage) to make better human network connections on the Web.

SUMMARY OF THE INVENTION

The present brokering scheme includes a Web-based social networking tool for online communities, professional networks, and corporate intranets. Social networks are networks of people connected by trust, shared values, and mutual need for cooperation. Social communities, cooperative business relationships, and professional associations are all examples of social networks. The present system creates social networks to find partners, clients and people with shared interests and values. This system is also used to share knowledge, build and strengthen communities, build teams, and map and analyze complex organizational networks.

Information networking tools now available on the Internet are inadequate to serve the needs of social networks. Information networking tools, including tools for "knowledge management" currently look for relationships between words. But social networking tools have to also reveal relationships between people in order to provide real value. This is why a directory of people and their expertise is generally not enough to evaluate a potential relationship. For most important relationships -- potential partners, significant clients and suppliers, etc. -- more information is needed to establish trust, mutual values, and other forms of compatibility. And, yet, since some of this information is personal or proprietary, social networking tools also have to insure privacy and protection.

This is the major challenge: How can people reveal enough about themselves so that the right people can find and evaluate them, while preventing the wrong people from accessing that information? This seems to be a contradiction, and until now has simply not been possible.

The present system resolves this apparent contradiction with a unique design that combines extensive knowledge of social networks with sophisticated software agents.

Working together, the present tools provide for rich profiling, access control, verification, and network brokering assure that privacy will not be compromised and that matches will be accurate and valuable to both parties. The present methods for network brokering are unique and are a major requirement for an effective social networking tool.

The present network broker works in two ways. First, it works by emulating a human broker. Human brokers are given confidential information by multiple parties because they are trusted to never reveal the information without permission of the owner. Likewise, the present network broker can access confidential information and use it to make highly accurate matches, but will only reveal the information when given permission by each user's personal access agent.

Second, the network broker can follow links based on trust, in the same way that people do when they are "networking" – except much more quickly and efficiently. When people network, they naturally start with the people they trust and ask for trusted referrals. Often they then also talk to the referrals and ask them for trusted referrals. Each step in the process involves a link of trust, and so the process is much safer than simply looking in a directory. The present networking system can semi-automate this process, and also speed it up tremendously. Other networking systems currently available through the Internet have portions of this process available, but lack critical features to make the process really work, including rich profiles, personal access control, and a sufficiently sophisticated network broker. The present scheme combines all of these features, all of which are required for effective social networking.

For example, the present scheme allows for rich profiles that can include detailed data on professional capabilities, history and accomplishments, goals, current projects, and professional networks, and also information about personal background, interests, values,

goals and networks. Making successful matches often requires both personal and professional compatibility and high degrees of trust. Tools which lack rich profiles usually provide poor matches and require too much work, and risk, from the users to do extensive additional evaluation.

The present scheme also includes a method for profiling personal networks that is used to guide the network broker in making accurate matches even when the user has not yet entered specific contacts. Profiling a person's networks gives the present Network Broker the ability to find likely connectors to targets, even when the targets themselves are not yet listed in the system. Profiling a person's networks also gives people considerably more information that they will need to evaluate matches that are returned. Knowing what kinds of networks a person has is often as important as knowing what kinds of capabilities or interests they have.

Also included is a Personal Profile Building Agent that guides users in the process of building profiles that are most effective related to their objectives. The present method also makes it more likely that people will create profiles that will be useful to themselves and others, by making it easy for them to incrementally add to their profiles based on their immediate and changing objectives and personal styles of working.

Personal access controls that put control over access to personal information in the hands of the user, not a system administrator, are also provided. In other systems, access control is usually controlled by a system administrator for the purpose of protecting the organization, not for protecting individual users. The present brokering system may utilize a set of default security values for different profile sections. Users can then adjust these defaults before they begin to complete their profile, and also as they build and edit the profile. The users are also able to give a security value to a particular detail within a profile

section that will override the default section security value. In addition, users are able to customize the default security values by adding additional conditions or rules to them. The end result is that different sections of a person's profile will have different security codes and/or rules attached to them, making it easy to share only certain parts of the profile with certain types of people.

This system of security values is matched by a unique method of applying Access codes to individuals, organizations, or groups defined by customizable sets of criteria. People are able to access any parts of a user's profile or other protected information that have a Security value equal to or lower than the Access Code that the user has assigned to them. The personal access control agent is also able to autonomously determine the level of access to give a party previously unknown to it. Other systems require that all parties, or their organizations, be specifically identified with an access level in advance.

In addition, the present personal access control agent is also able to autonomously interact with other agents, including the network broker, and with other users or their agents, via the network broker agent. Other systems of access control lack this type of autonomous interaction.

Thus, the present system includes:

1. A Network Broker Agent that can

a) Use confidential information to make matches but which follows the instructions of personal access agents regarding release of confidential information, and

b) Follow multiple links of trust to find trusted connections to desired targets.

c) Look for and use likely connectors to targets, making use of personal network profiles as well as knowledge which it accumulates regarding likely connectors based on previous successful matches.

2. A Network Verification Agent that can

a) Verify user-supplied profile information by automatically checking references,

b) Apply techniques of social network analysis to add extra measures of verification, and

c) Be used to supplement human methods of verification.

5

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example, and not limitation, in the figures of the accompanying drawings in which like reference numerals refer to similar elements and in which:

Figure 1 illustrates an example of a client-server environment within which the present brokering system may operate.

DETAILED DESCRIPTION

A scheme for brokering human networks over a computer network that may use a client-server paradigm is disclosed herein. Although discussed with reference to certain illustrated embodiments, upon review of this specification, those of ordinary skill in the art will recognize that the present scheme may find application in a variety of systems. Therefore, in the following description the illustrated embodiments should be regarded as exemplary only and should not be deemed to be limiting in scope.

As indicated above, human networks are central to many (if not all) value-creating activities and operate at multiple levels. Within and using the present brokering scheme, users are able to:

- Quickly narrow the choices in finding and evaluating new partners, clients, colleagues, suppliers, employers, employees and information sources by learning enough about them to properly evaluate their potential.
- Get recommendations and introductions from a trusted source to build extended networks based on trust and value.
- Screen incoming information and requests and maintain privacy when desired.
- Connect with more of the right people and clients to foster new relationships

The present brokering scheme offers users a rich environment to create and maintain all types of human networks supported by online interactions. When used in a client-server paradigm, the client-side tool may be embodied as a Java applet that works within a web browser (e.g., Microsoft's Internet Explorer™ and/or Netscape's Navigator™), thus offering a rich profiling environment for sharing information. The corresponding Web application server may then be made up of gatekeeper and

network broker intelligent agents that are trainable to emulate the functions of a human broker, making highly accurate matches while protecting personal privacy.

Introduction

The present scheme operates as follows: Initially, users create rich profiles needed for greater visibility and more accurate matching and evaluation of new connections. These profiles may be both locally stored (e.g., on a user's personal computer) and/or uploaded on one or more application servers that may reside on an intranet or a Web server for an online community. Users instruct a personal gatekeeper agent to control access to their own personal profiles and a personal search agent to find desired connections. A network broker operating on the server(s) is then used to evaluate matches and broker relationships. All parties can trust the broker to follow their instructions regarding the match desired and the degree of privacy desired.

This scheme offers a method and framework for matching people's interests and building personal connections to the right people and information they value. Getting the right person's attention is what really matters and is a necessity for creating almost any type of value. However, in the real world, the factors that influence individual and group attention are very complex. The sophisticated software agents employed as part of the present brokering scheme build on the complexity of human relationships in ways that are transparent to the user.

The brokering scheme also addresses the flip-side of attention by protecting personal privacy. Maintaining user privacy is an enormous issue on the Internet and is likely to continue to grow in importance as more and more people conduct online transactions. The brokering scheme addresses privacy concerns by managing access to attention because the

scheme is based on the understanding that the more visible it is possible for a user to be, the more it becomes necessary for that user to be selectively visible.

To better understand the present brokering scheme, consider that different embodiments thereof include one or more of the following components:

1. **Profile Builder**

A tool (e.g., a software application or component) that helps people and organizations (collectively, "users") reveal their capabilities, projects, goals and values so that others can adequately evaluate them. The present profile builder is rich and customizable, asking people what they want to accomplish, and then helping them build a profile that can accomplish those goals. Each individual has total control over what goes into his or her profile, who has access to it, and what networks to connect it to. This is significant because in order to make good connections it is as important for others to evaluate a user as it is for that user to evaluate them.

2. **Personal Gatekeeper Agent**

This tools allows people to protect the information in their profiles and their attention from inappropriate access, and make these personal profiles connectable. Is easy to use and is customizable, allowing access categories that can be assigned at either macro or micro levels within one's profile. The Gatekeeper Agent may also be trainable – learning from both direct instruction by its owner and by "observing" its owner's behavior.

3. **Personal Search Agent**

A tool that guides the user in constructing a profile for a search target.

4. **Network Broker Agent**

A network agent that emulates the function of a human broker negotiating between users' personal search and gatekeeper agents.

4. **Verification Agent and Network**

A network agent that authenticates and verifies information people have recorded in their profiles. This agent may work in conjunction with third-party systems such as those available from/through TRUSTe, 1180 Coleman Avenue, Suite 202 San Jose, CA 95110.

The Verification Network makes use of features of the present system to verify user profiles. This simplifies and strengthens the verification process for people offering their services (e.g., job seekers or consultants), for people seeking services (e.g., employers or prospective clients), and also for people who verify others (references given by job seekers or consultants). The Verification Network can be explained as follows:

Virtually every professional in the world uses a resume to profile his or her capabilities and to list references. Many professionals and employers are now using Web services to post and locate resumes. The present Verification Network adds enormous value to this well accepted and universal practice and in the process introduces new users to other features of the brokering system which they can continue to use long after their initial purpose is fulfilled.

Currently, individuals often list three or more references on their resume and any prospective employer or client who wants to evaluate the resume must personally call each reference and do other forms of due-diligence. Further, each of the references listed has to give the same (or substantially similar) evaluation to each organization that is checking references in order to be of any value. The present Verification Network automates the process of reference checking to eliminate the need for each employer or potential client to call individual references and for each reference to provide the same verification multiple times. In this system the verifier provides a verifying authority with a single verification which then becomes accessible to any interested parties.

In one embodiment, users/subscribers enter their profiles into the system in the fashion described below. This can range from a traditional resume that they already have to use of the system client (see below) to build a full user profile. At the same time, these users identify people who can verify their profile (or resume) in general and also, if desired, who can verify specific capabilities, accomplishments, or values.

The server communicates with these verifiers by email or other ways and asks them to verify the selected profile or profile sections. The server receives and evaluates responses from verifiers, and records the results in a special verification section of the profile which can only be edited by the server. A copy is also retained on the server, which will be compared with the copy on the client.

When other users – and their agents – want to evaluate a prospective connection, they may access verification information for the individual or company being evaluated. At minimum they can access a verification rating. If the user's Gatekeeper grants them additional access they will be able to access more details of the verification.

Users are able to link their profiles to multiple verifying and connecting servers. Profiles are thus fully portable, and are not locked in to a single Web site or organization.

An Example

With the above background, it seems now appropriate to explain further details of the present brokering scheme by presenting an example of how it might be used. In the following scenario, a user, Julia, needs to make new contacts to help with a project; her problem is how to make those contacts. Making the right new people connections involves three basic steps, each sometimes more difficult than the prior one:

1) Getting a list of candidates.

2) Selecting the most qualified candidates – the ones with capabilities and experience more finely tuned to the user's needs (we'll call these targets).

3) Getting a target's attention.

The first step (assembling a list of candidates) may be done readily with a good directory or database; but that's about where such tools end. The present brokering scheme addresses the more difficult steps.

Our exemplary user, Julia, is a senior consultant in a mid-sized consulting firm in New York. The company has offices in Chicago and Los Angeles, and close alliances with other companies in London, Brussels, and Tokyo. Julia has targeted as a prospect an international company that she had never dealt with before. To make any progress, she knows she needs access to that potential client in the form of a high level introduction, and probably also some inside information about the prospect's needs for services. She had sent a message and material and made a follow-up call to the prospect, but the logical contact there

was overwhelmed with work and other requests for his attention. He took a superficial look at her materials, said "Sorry, I'm too busy to respond." and then didn't want to talk anymore.

Prior to the present brokering system, Julia may have been at a loss for further solutions to penetrate the prospective client. At best, her option might have been to go through her address book and find people she knew who might give her a lead, call each of them in turn, each time explaining her needs, and then follow-up on any leads, which might generate other leads, as well as many blind alleys. Sending a broadcast email or posting a message on the company's discussion server might have helped, but too often such attempted solutions either return little information, or leads that didn't really match Julia's present needs.

With the present brokering scheme, hover, Julia can explain her needs once (to her personal Search Agent) and then get the best leads available, without depending on getting her colleagues attention or full understanding of what she was looking for. If just one of the individuals in the company's alliance network has a good connection to Julia's potential client, the present brokering scheme offers an excellent chance of finding it.

Assume for purposes of this example that Julia's company's network had only recently deployed the present brokering system, so people within the company may still be adding information and contacts. Fortunately, the present system does not require that all members of a network be equally thorough in entering information. In this case, suppose that 3/4 of the company's workforce had at least profiled themselves and their networks and entered a few specific contacts. This may lead to, say, 10,000 contacts in the network, of which perhaps some 8,000 or so may be unduplicated.

This is not a vast number of contacts, however, as more and more members of the company enter their information this number will grow. Further, extended networks may

also be included to encompass outside contact lists as well. As a result, a total number of contacts in an extended network could easily exceed 500,000 or more, with the potential to grow even larger. Nevertheless, even the 8000 unduplicated, high value, high access contacts is substantial material for the present brokering scheme to work with, especially since the network profiles contributed by participating members also provide considerable information needed to point to likely connectors to most targets.

To perform her search, Julia need only do some or all of the following:

1. Enter a summary description of her search (i.e., who/what she wants to connect with, for what purpose, etc.).

2. Profile the target. In this case Julia had the target's name (her prospective client company) and also entered a profile including other important criteria: location, industry, specific type of company and products.

3. Profile particular target contacts. Here, names of people, roles, and other factors like professional interests that would increase the likelihood of a good match.

4. Profile likely connectors to the target. For example, specific or general types of members of the target's network: clients, suppliers, partners, advisors, or other associates.

5. Indicate level and types of connection/verification desired. In this case Julia may specify that she wants only close, trusted contacts, and a maximum of two levels of connections.

6. Specify desired results: Julia may want both connectors to this target as well as to other potential targets.

7. Launch the search (and also save it for future use in similar situations).

The brokering scheme may then implement the search on the alliance network server.

From this search, assume the following results were obtained:

1. Actual connectors to this target. Jane Adams, Chicago Office, was identified as having a close connection to a Vice President in the target organization.

2. Likely connectors to this or similar targets. Names of four people in the network who have close contacts in this industry and with this type of company, or with likely clients of this type of company.

3. Messages from connectors. "Call Jane before you call her contact."

Presumably, Julia will be quite happy with these results, especially since developing this list did not take that much of her (or anyone else's) time. With these results she could now make just a few phone calls to exactly the right people.

Architecture Overview

The present brokering system is a multi-tier system, which in one embodiment includes some or all of the following:

- A Java applet client (e.g., which can be downloaded from a server) that resides on a user's local system. The client contains a local database as well as client-side agents of the following types:
  - Profile Builder agent
  - Gatekeeper agent
  - Search target profiling agent
- An HTTP server,
- A Java application server (which can be combined with the HTTP server), which includes the following agent applications:
  - Search agents
  - Gatekeeper agents
  - Network Broker agents
  - Verification agents

- A database server, containing a secure, composite data structure which maintains information about all users of the system plus a record of prior searches and matches which the Network Broker agent can access to learn and reapply successful search strategies.

Many components of the present system may be embodied as Java applets to maintain the richness of an object-oriented approach while using a conventional web browser and HTML (hypertext mark-up language) as the delivery platform for the user interface. For compatibility with Java-based interface agents that might be delivered through the user interface, a Java application server should dynamically generate the HTML ("compiled HTML"). Database storage is preferably (though not necessarily) also object-oriented rather than relational. There should be a high capacity database on the server, and more limited "persistent store" capabilities on the client-side.

The client application may be distributed as a package containing the items listed below, to installed in a registry (or similar) portion of the user's personal computer (or other Web-capable appliance) operating system, and stored in a named directory:

- A Java applet or applets
- Persistent store for the applets
- A web browser (installation optional if the user already has a browser which supports the current version of the Java virtual machine).

The locally stored applets conduct a dialog with the user to build profiles off-line. Data resulting from this dialog is stored in a local database (e.g., a portion of persistent memory of the user's personal computer). This database may be encrypted or otherwise secured against prying eyes and file theft, since client machines in any organization are much

more loosely guarded (if at all) than enterprise-level servers. It is believed that an object oriented database will be important for both the client and server due to it's superior integration with the object design of the present brokering scheme and due to the extreme complexity and flexibility of matches and links between objects that will be required. However, in some cases, other database designs (e.g., relational databases) may be used. A "local gatekeeper" ensures that only data that the user has designated to be shared gets sent to the storage on the server. Search queries may also prepared off-line, before being sent to interact with the server's broker agent.

As demonstrated by the above example, it is important for the user interface to respond to differences in user objectives, style and context, and to changes in these factors over time. To accomplish this, users will be queried on their objectives and preferences in their initial session with the broker client (e.g., in an "Orientation Interview" conducted by/through the user interface) and in subsequent sessions. Advanced functionality may also include monitoring a user's behavior to detect style and preferences. Information about user objectives and preferences may also be stored in the local database, along with the user's profile completion status, information about how the user has used the brokering system in the past and results of use, and possibly also user satisfaction with results.

Information about the user's objectives and context (e.g., industry or profession, country, etc.) may be used to select and customize prompts to present to the user, and (in some embodiments) to make suggestions or offer information to the user. For example, information collected on the user's organization type and profession may be used to select prompt variants. Information on the user's objectives may be used to guide the user to complete sections of the profile that will be most necessary for achieving those objectives, and to optionally skip sections that are not essential for his/her objectives. Information on

the user's completion status may also be used, alone or along with user objectives, to guide the user to complete the next most important section when s/he logs in next. User completion status may also be used to reward users for profile completion and for the value that this provides to other users.

The interface may also respond to differences in people's style and preferences. For example, a task-focused person who wants to cut to the chase and solve an immediate problem may get a more minimal set of initial questions than, say, a more curious or expressive person who wants to carefully construct his/her profile. In the case of the quick-start person, the interface may cut the orientation and profile building to a minimum and quickly find what the user wants to do. For example, if the user wants to find a particular type of contact, instead of asking him/her a lot of questions about himself/herself, a "search deva" (search profiling agent) may ask the user simply to profile the person s/he wants to contact. This will get the user familiar with the basic elements of a profile. Then the deva may remind the user that that person s/he is looking for will probably need to know some things about him/her. This will give the interface a task-focused reason to get the user to start profiling himself/herself. Then, in later sessions, the interface can gradually get the quick-start user to fill out more of his/her own profile. For example, when results come back from a search and a good prospect wants to know more about the user, then the user will have to reveal more information in order to complete that connection.

This type of personalization generally requires an intelligent, dynamic and non-linear interface. The present brokering systems includes software tools that contain the intelligence (rules, etc.) needed to respond to user preferences, context, and objectives. Since each question presented to the user in each profile section will be an object, the interface may dynamically select which questions and prompts to display, how to number them, etc., based

on accumulated information stored in the local database. A basic object hierarchy for the client application is shown below.

InterfaceToUser

This object may contain the intelligence described above and also the interface objects needed to feed the local database.

LocalDB

Local database. Sections of the database include:

UserStatus (Described above as information regarding the user profile, etc.)

OrgProfile The organization profile may be completed once for each organization and may be accessible to each user's client in that organization. This profile may be completed by a single person or contributed to by multiple people. Some information included in the profile may be dynamically added by the brokering system itself, based on collective responses of organization members. Sections of the database may include:

Capabilities, History, Values (e.g., Culture, and Basic_Values), Goals, Projects (e.g., collected from member profiles, can also be added by a system administrator), Networks (e.g., collected from member profiles, can also be added by a system administrator; subcategories may include Profile_Of_Networks, Contacts (both Internal and External), and Resources)

PersonalProfile The sections of the PersonalProfile may be very similar to the OrgProfile. Both objects may inherit from an abstract Profile class. The sections of the PersonalProfile may include:

> Capabilities, History, Values (e.g., Interests, Style and Basic_Values), Goals, Projects, Networks (e.g., Profile_Of_Networks, Contacts, ContactProfiles (This database may store profiles of the user's contacts that were either entered by the user or downloaded from a server. If the profile was downloaded from a server, the profile may contain a link to the copy of the profile stored on the server so that the profile can be updated when it is accessed (as allowed by the profile owner's Gatekeeper Agent)), Gatekeeper (This database stores general and specialized Gatekeeper instructions and a log of Gatekeeper actions and user responses to these actions (e.g., satisfaction, correction, etc.). These will be used to train the Gatekeeper Agent. Note: The access and security codes (if any) used by the Gatekeeper Agent may be stored in the user's Profile), Searches (This database may store prior search parameters, results, and user responses to results so that searches may be reused, modified, and improved).

InterfaceToServer

This component will allow a user to upload profile sections and agent instructions to the server and download results and other communications from the server. A database

replication and file optimization scheme may also be included. The interface to the server should be closely connected to the client-side gatekeeper agent.

ClientGatekeeper

The client gatekeeper insures that data marked with an access code for "Self Only" will not be shared with the server. The client gatekeeper will also respond to requests by the server for information stored only in the local database, or for specialized responses to search results (e.g., requests for additional information or actions by the user). Advanced functionality in some embodiments may include the ability to filter all types of incoming information and requests for the user's attention, including email.

Having thus explored the client-side of the present system, the server-side thereof may be examined. More than one server may be used, for example a broker server and a database server may be separate entities, even if hosted on a common platform. The broker server may be a Java application server that includes the functions of an HTTP server, dynamically serving HTML content and associated Java applets to the client. It may also host Java applications that serve the functions of broker server-side agents, including the Network Broker agent, Gatekeeper Agents, and Verification Agent. It should, preferably, interface in a secure fashion with the database server.

Client applets may be stored on the Java application server and delivered to the client on demand in the context of HTML pages. Since these applets may also be stored on the client, the server should query the client for to find out if the most current version of the applet is present on the client and, if not, provide it.

A Java application communicates with associated data structures when the client gives it new information to store there, and updates the server-side gatekeeper for that user with new instructions for maintaining secure access. The Java server also accepts search agent instructions, and communicates these to the broker agent. Upon a successful match to the criteria of the search, the server communicates back to the client regarding the successful path to the repository of information or contacts that satisfy the search. It is desirable that the server be able to request new information (not currently stored on the server) from the clients to see if a match is possible based on information not yet shared; this could conceivably lead to human intervention and negotiation towards selective release of the information.

The various applets may be stored in directories on the server, individually and as packages to be downloaded to new users. Most of the server-side data structures may be stored on the database server. In one embodiment, the server components include:

1. HTTP Server
2. Interface-To-DB Server
3. Interface-To-External Servers
4. System Agents:
    a. Network Broker Agent. The Network Broker Agent should search the User Profile database to look for matches against the criteria specified in search parameters sent by a client. It then should evaluate the closeness of fit to the search parameters. If the search parameters specify connection criteria, such as level of trust, type of connection, etc., then the Broker Agent may have to discover and evaluate connection paths between the searcher and the prospective target. For each prospective target found, the Broker Agent next should ask and receive permission from the target's

personal server-side Gatekeeper Agent for release of requested information, which is then sent back to the requesting client.

b. Personal Gatekeeper Agent. The server-side Personal Gatekeeper Agent evaluates any request delivered by the Network Broker Agent and determines what information may be released. This will be the case in response to both searches and browsing functions initiated by clients. The server-side Gatekeeper Agent may also request any additional information it needs in order to better evaluate what access level to give to the request.

c. Network Verification Agent. The Network Verification Agent should respond to any updated verifier information sent by a client. The Verification Agent may send an email message to each verifier listed, asking the verifier to confirm the client-supplied information to be verified. The Verification Agent may also receive replies to these emails and evaluate them. Finally, the Verification Agent may place a "verification stamp" in a section of the requesting client's profile containing the results of the evaluation. This verification stamp should be editable only by the Verification Agent and not by the user or any other entity or application.

The database server may include some or all of the following components:

1. UserDB. This database may store all the profiles and instructions uploaded by system users. Database sections may include: User_Profiles, Gatekeeper_Instructions, and Search_Instructions

2. SearchResults. This database may store results of searches to be used by the Network Broker Agent in refining and reapplying any successful search strategies. Personal Search Profiling Agents operating on the clients may also access this database in order to recommend search strategies to the users and prompt for information needed by these strategies.

3. ExternalServerIndex. This may be a database that will help a server extend searches to external servers. As the number and locations of servers proliferate, a system to index all profiles on the extended system of servers may be useful in order to direct extended searches from one server to another.

Use Cases

Although the above example of a user, Julia, searching for a contact provided some details of how the present brokering system operates, a more general case may also be helpful. Through their individual client applications, users can enter plain language descriptions (as opposed to complex SQL (structured query language) or other database-specific descriptors) of their Search Nature. This can be used for a human readable description of the search – by prospective targets, connectors, or human brokers helping with the process. In some embodiments, machine parsing and comprehension of these search terms may be provided.

Users may also enter a pre-selected category describing the search by specifying an Action/Object pair, e.g., "Find Partners" or "Offer Services", or "Exchange Ideas". These categories can be used to help optimize search results and also to group searches for refining search strategies, and to retrieve stored searches. In addition, compensation requirements, if any exist, can also be specified. This includes compensation the user is willing to pay to targets or connectors or compensation that the user expects if offering services or goods.

Users also profile the target organization and target person to specify the skills, capabilities, and values that are being sought. The interface for specifying such information may include a place to indicate the relative importance of any profile criteria specified – e.g., required, very important, important, etc. Users may also profiles likely connectors to the target. This is optional but can be important if the type of connection to the target is important, for example, if the user wants a trusted recommendation and introduction to the target.

Some embodiments may include a Search Profiling Agent that may help the user specify the kinds of profile information that are most likely to result in matches for the desired search objective. For example, the Profiling Agent may be able to recommend what type of connectors to look for, or what kinds of information the target is likely to need in order to respond. If the user is in a hurry and leaves out important sections, the Search Profiling Agent may give the user reasons for completing those sections and ask them to do so.

Verification parameters, if any exist, may also be specified. The Verification instructions will help the broker agent plan its search path (i.e., does the user want to limit the search to:

His/Her own closest connections?

The closest connections of his/her closest connections?

The people in his/her organization?

People in allied organizations?

Connections of people in allied organizations?

The global universe accessible by the brokering system?

Only verified or certified targets?

etc.)

A search may also be initiated or saved for later use whenever the user adds or modifies his/her own profile information regarding projects, goals, or interests. Projects especially relate to searches if the project has requirements that are not yet filled. After entering project information, user may be prompted, "Do you want to start a search for project requirements?" If yes, the project specifications will be used by the agent as the basis of a search and the user may be prompted for additional instructions needed to carry out the search. Likewise the user can initiate a search for people who share common interests, values, goals, or background.

Ultimately, a user launches the search. The search is preferably first launched on the user's local system to look for matches or likely connectors among the user's own locally listed contacts (including locally listed contacts of other members of the user's organization); next the search is uploaded to the server for a more extended search.

The server receives search instructions from a client and instantiates a Broker Agent to carry out the search. The Broker Agent receives search parameters (e.g., through its interface to the server) and responds with search results. To do so, the Broker Agent parses the search into component parameters thereof and conducts a search in the User Profile Database located on the server to try to find best matches. If the search parameters include

connection types and strengths, then the Broker Agent will seek connection paths that match connection parameters. This may include strategies such as: starting with targets and working backward to try to connect to the searcher, via likely connectors; or starting with the searcher's contacts and working outward to try to connect to targets or other likely connectors.

Once matches to search parameters are found and evaluated, the Broker Agent will start with the best targets and likely connectors and negotiate with the target's Gatekeeper Agent for release of information about the target to the searcher. The target's Gatekeeper Agent will evaluate information which the searcher has allowed the Broker Agent to reveal in order to determine what level of access to assign to the searcher's request. Based on the access code assigned to the searcher by the Gatekeeper, the Gatekeeper will give the Broker Agent permission to report back to the searcher any requested information that has a security code equal to or lower than the searcher's assigned access code. In some cases this will be all information requested, in other cases it may include some information but not include specific contact information (name, etc.); in still other cases it may be no information.

If the Gatekeeper is interested in the request but cannot assign a high enough access code to the searcher to release the information requested, the Gatekeeper Agent may (if previously instructed by its owner) ask the Broker Agent to query the searcher for the additional information that it needs to release the requested information. This request for more information will then be relayed to the searcher's server-side Gatekeeper, which will decide what to do with the request. For example, the searcher's server-side Gatekeeper may a) supply the information, b) deny the information, c) send the request back to the searcher's client to request action of the searcher's client-side Gatekeeper or directly of the searcher

him/her-self. Such additional information supplied by the searcher may then be relayed back to the target's Gatekeeper for re-evaluation of access.

If the target's Gatekeeper responds negatively (or sub-optimally) to a searcher's request for information, the Broker Agent may then attempt to find a trusted connection path between the searcher and the target (if it has not already done so). If a trusted connection path is found, then the Broker Agent will submit this additional information to the target's Gatekeeper Agent to try to improve the access assigned. When the Broker Agent is looking for likely connectors to targets, the Broker Agent will be asking the connector's Gatekeeper Agent for permission to search the connector's contacts for targets or other likely connectors. This will allow the Broker Agent to conduct extended searches through multiple "degrees" of connection.

Once results are obtained from target or connector Gatekeeper Agents, the Broker Agent will collect all results obtained and rank and report them back to the searcher. The report back to the searcher may include some or all of the following:

Direct Hits

1. A ranked list of "direct hits" (people, organizations, information that matches the search target, etc.).
2. Hyperlinks to all relevant evaluation information that is accessible to the searcher.

Connectors

1. A ranked list of potential connectors to the target. The highest ranked connectors will usually be people who are most likely to have the strongest connections to direct hits and who also have strongest connections to the searcher.

2. Hyperlinks to all relevant evaluation information about connectors that is accessible to the searcher.

Messages and Requests

1. Messages from potential targets or connectors, such as, "Please contact me personally for more information--or for a personal introduction."

2. Requests from potential targets or connectors asking for more information required to permit more complete access

Because searches can be quite complex and because User Profiles will often contain varying degrees of information related to desired parameters, search strategies especially suited to these conditions may need to include:

a. 7G adaptive network (neural network) technology to evaluate and optimize complex, fuzzy matches, and which can make use of weighted connections within search paths.

b. Advanced 7G capabilities.

c. Modification and use of other third-party search engines. For example, the broker system may parse and store user profile parameters in text files which can then be grouped according to parameter category and individually indexed and searched using available search tools. This will help improve the accuracy of matches by constraining searches to desired categories.

d. Development of custom built object-oriented search strategies and technology.

Algorithmic Details:

I. Object Model Overview:

Client contains (in addition to PersonalProfile, etc.):

ClientSearchDeva – which gets search parameters from UI, conducts searches on local database (db), and assembles SearchInstructions for sending to Server

InterfaceToServer – which sends and receives messages, objects and data to the server. This object interacts with UI and local db and with other Client objects such as PersonalSearchDeva.

Server contains:

InterfaceToUser – which receives input from users, including searches, and sends results and messages to users, including search results. In the case of an incoming search, this object will send a message to the NetworkBroker class to cause it to instantiate a new NetworkBroker object to carry out the search.

NetworkBroker -- this is a class instantiated to handle each search received. Thus, there will be multiple NetworkBroker objects active at any given time.

SearchDatabase – containing temporarily stored or archived searches and results.

Various profile and contact databases.

Database storing access deva instructions.

NetworkBroker objects contain:

SearchInstructions (received from Client by InterfaceToServer and copied to this new NetworkBroker)

TargetFinder – will conduct a simple search to match query and find all possible targets. Will create **BrokerResults** data object

TargetEvaluator – an object uses BrokerResults to implements access deva instructions for each target found. It modifies BrokerResults based on evaluation. It will also have other functions in full searchers – e.g., evaluating and scoring weighted searches. (Note –may be implemented as multiple, threaded TargetEvaluators, one for each target.)

BrokerResults data object (a private object only known to Network Broker)

ResultsToSearcher data object (sent back to client)

II. Detail of Objects and Behaviors

Client

Browse UI

Get user input for query.

ClientSearchDeva (a system interaction object in the **NetDevaClient**)

Assemble **SearchInstructions** from Browse UI – user input.

SearchInstructions will include:

Search Description (if provided by user)

SearchID

Type of search (e.g., a browse query, or anonymous, weighted search).

Query conditions to match.

Whether search is local only or also to be sent to Server.

If sending search to server, whether to:

a) Send instructions now and ask for immediate search while client is online.

b) Whether to send instructions now and notify when results are found (user intends to be offline while search is occurring)

c) Whether to store SearchInstructions and upload when next online.

d) What data to return with results:

i) Send only Target names and Link to data on server.

ii) Send names, Link, and additional data:

Data matching query conditions

Email, phone, address

Other profile data as requested.

Carry out SearchInstructions

Find targets in local db

Send SearchInstructions to Server (if applicable)

Server

InterfaceToClient

Receive and acknowledge **SearchInstructions** and initiate a **NetworkBroker** to carry out match on Server.

NetworkBroker (instance for a particular search)

TargetFinder

Find targets among other users on server,

add Link ID to **BrokerResults.TargetList**

Find targets among contacts of other users on server

Add Link ID to **BrokerResults.ContactList**

Find Connector matches among users and contacts on server.

Add Link ID to **BrokerResults.ConnectorList** (Not for POC)

Note: **BrokerResults** is not shared with objects outside of the **NetworkBroker** assigned to this search.

TargetFinder object will thus include methods to read SearchInstructions and do the above steps, plus data members of BrokerResults (which will include the three lists, TargetList, ContactList, and ConnectorList).

TargetEvaluator

Start with **BrokerResults.TargetList,**

For each target user in the list:

i. Create a Target object in **ResultsToSearcher.TargetList**.

This will be the search result data object sent to the Searcher.

ii. Check Security code for requested data, e.g., name, phone, email, location, data matching query, and any other profile information requested.

iii. If all data requested is "Public", add data to Target in TargetList.

iv. If any data is not "Public", check for AccessCode assigned by Target to Searcher.

v. If AccessCode is not found for Searcher, check for AccessCode for Searcher's Organization (employer or membership associations). Optionally, provide ability to add contact detail for organizations including access and security codes. Or, check to see if Searcher is in same organization as Target's organization. If so, give Searcher Access Code of "Medium" (where "Medium" Security code is defined as "In my organization"). Otherwise give Searcher AccessCode of "Public".

Note that Searcher's name and organization name may need to be part of SearchInstructions in order to do lookup in Target's contact access code. In some cases a more stringent way of verifying that the Searcher or Searcher's organization matches a Target contact may be needed. Note also that Target will not learn Searcher's name for this procedure since the lookup is done by Network Broker. However, in other embodiments targets may need to know when they've been browsed or searched plus any profile information the Searcher

wishes to share with targets, though not necessarily including the name of the Searcher if Searcher wishes to remain anonymous.

vi. Add SearcherAccessCode found to Target in TargetList. If no SearcherAccessCode found, Target.SearcherAccessCode = null.

vii. Add all data to Target in ResultsToSearcher.TargetList where SearcherAccessCode >= SecurityCode.

viii. If the Target's name is not accessible (not "Public" or has a Security Code higher than AccessCode assigned to Searcher) then substitute "Person N" for the Target's name (where "N" is a sequence number, like "A", "B", etc.) Post POC we should also not use an actual ID as a link to the user profile on the server, but instead create a temporary link ID that will be stored in the TargetList. The actual ID will also be stored there so that the Broker can look up the actual ID when later given the temporary ID by the Client.

ix. Repeat these steps for each target in ContactList and ConnectorList. (If desired, find connectors.)

InterfaceToUser

Send **ResultsToSearcher** (containing SearchID, TargetList, ContactList and ConnectorList) back to Searcher's NetDevaClient.

In some embodiments, BrokerResults may be stored in memory while a user is online in order to respond to user requests to browse detail for particular targets. In other embodiments, after Searcher is offline, BrokerResults may be stored along with an expiration date, after which it will be purged.

BrokerResults contains **Target** objects which contain:

SearcherID

SearchID (matching the ID for the Search on the NetDevaClient)

Link ID to target (user or contact) on server.

Temporary Link ID sent with ResultsToSearcher to Searcher

SearcherAccessCode assigned by target to Searcher

Client

InterfaceToServer

Receives ResultsToSearcher as **ResultsFromServer**. Matches SearchID to SearchIDs in Client. This will indicate which search the results are in response to, including whether results are for a browse or a search action so that client can implement it correctly.

For Browse results from server:

If query in Selection Criteria table is the same as the query for the results (match SearchID in client to SearchID in ResultsFromServer), then display the results of the query in the browse table. (Note: When a user creates a new query the browse table will always be cleared until results are displayed.) If user wants to view detail for a result, lookup the user or contact in the local db or server, as appropriate, and display accessible data.

Optionally: If query in Selection Criteria table is no longer the same as the query for the results, then display message telling user that results for the query (using query description if available) have been received from server, and ask if want to view results now or later. If now, then clear the browse table and display new query in selection criteria table and "contacts of" list, and display results in contact browse table. (First ask if want to save existing query (and/or save it and put it in a LastQuery buffer.)

Search Strategies

I. Search for Targets – not considering trust links

1. Search on each criterion. Examples: Narrow overall search by searching first on required criteria, such as location, capabilities, or other. Search in capability section of profiles to match on capabilities. Narrow searches by broad industry and function groups. Search in network profiles section to match on type of clients, and other

network types. Search in history/education section to match on education criteria. Search in interests/values section to match on these criteria, etc. Searches may include both searches of structured fields and indexed text searches in text fields or "tagged" text files or tagged sections of text files (e.g., sections of resume text files that are tagged as "capabilities", etc.). Some text fields used in profiles may contain unstructured text fields, however the scope of these fields should be defined, such as "Current Skills." They may also be defined by association with structured fields indicating, for example, particular industries or functions.

2. Evaluate and record strength of match for each criteria.
3. Group matches by target (organization or person).
4. Combine matches into one "record" or object per target, giving strength of match on each criterion. (Note, steps 1 – 4 may of course all be combined in one operation per target.)
5. Apply overall match ranking algorithm or method. Method should be able to accommodate for missing or fuzzy data. For example, results of #4 can be input for neural network match ranker.

II. Discover Trust Links to Targets and Likely Connectors

This could be done as part of Process I if minimum trust links are specified as part of the search requirements. However, even in that case it may be desirable to use Step I as a way to narrow the universe of possible targets before the more resource intensive process of discovering trust links.

1. Start from searcher and follow links with minimum trust outward no more than x (say 3) degrees, or less if fewer degrees are specified in search parameters.

2. Start from targets identified in Process II and follow minimum trust links outward the same number of degrees. Note: Steps 1 and 2 will involve "extended searches" which may require Personal Access agent (Gatekeeper) approvals – i.e. to explore links of a connector. This will be required because in order to release information on link paths – i.e., you are linked to person A, who is linked to person B, who is linked to a target. The Broker Agent will ultimately need A and B's access agent approval for release of this information, so it would make sense to obtain it before following links of any link owner – otherwise, one can ignore link owners which do not want to participate in an extended search. In that case, the link will not be considered sufficiently strong by both parties and so would not be a good path anyway. However, it may be that the link owner has not gotten around to including the link requestor in his/her access instructions or that his/her access instructions are not very complete. In that case, it may be preferable to do the extended search first and then ask permission. Asking permission may involve getting the user's attention, which could take time and delay or fragment the search process. Better to do the search and then release info as permission is received.
3. Look for matches between potential link connectors found in steps 1 and 2.
4. Rank matches according to search path link criteria, e.g., strongest links, fewest degrees.

When searching for likely connectors, one can search all links available, if system resources and time permit. Or one can constrain a search by using a refined "likely connector" strategy that takes into consideration programmed and learned information about likely connectors, plus consultation of network profile section of potential connector profiles. These extended link search strategies will not only locate any known link paths in the system, but will also be used to identify "likely connectors" that don't directly link up to a target. Such likely connectors may indeed link up to targets, but available linking

information may be missing. Or, likely connectors may link up to targets who are not in the system. These connectors can then be contacted by the searcher (if permitted by the connector's access agent) for help in making connections "outside" of the brokering system.

When looking for likely connectors, some other possibilities are:

- Look for likely connectors to identified targets.
- Look for likely connectors to 1st and 2nd degree connectors to target's 1st and 2nd degree connectors.
- Look for likely connectors to the target profile.

III. Assemble & Evaluate Results Based on Process I and II

This process will combine weighted connection criteria with other target profile criteria. As stated above, this will be useful even when weighted connections are not required in the search parameters. First of all, having information on weighted connections will add value to the search results – value that the searcher may not have anticipated. Second, weighted connections – e.g., links of trust – may prove valuable for getting a target's (or connector's) access agent to release information to the searcher.

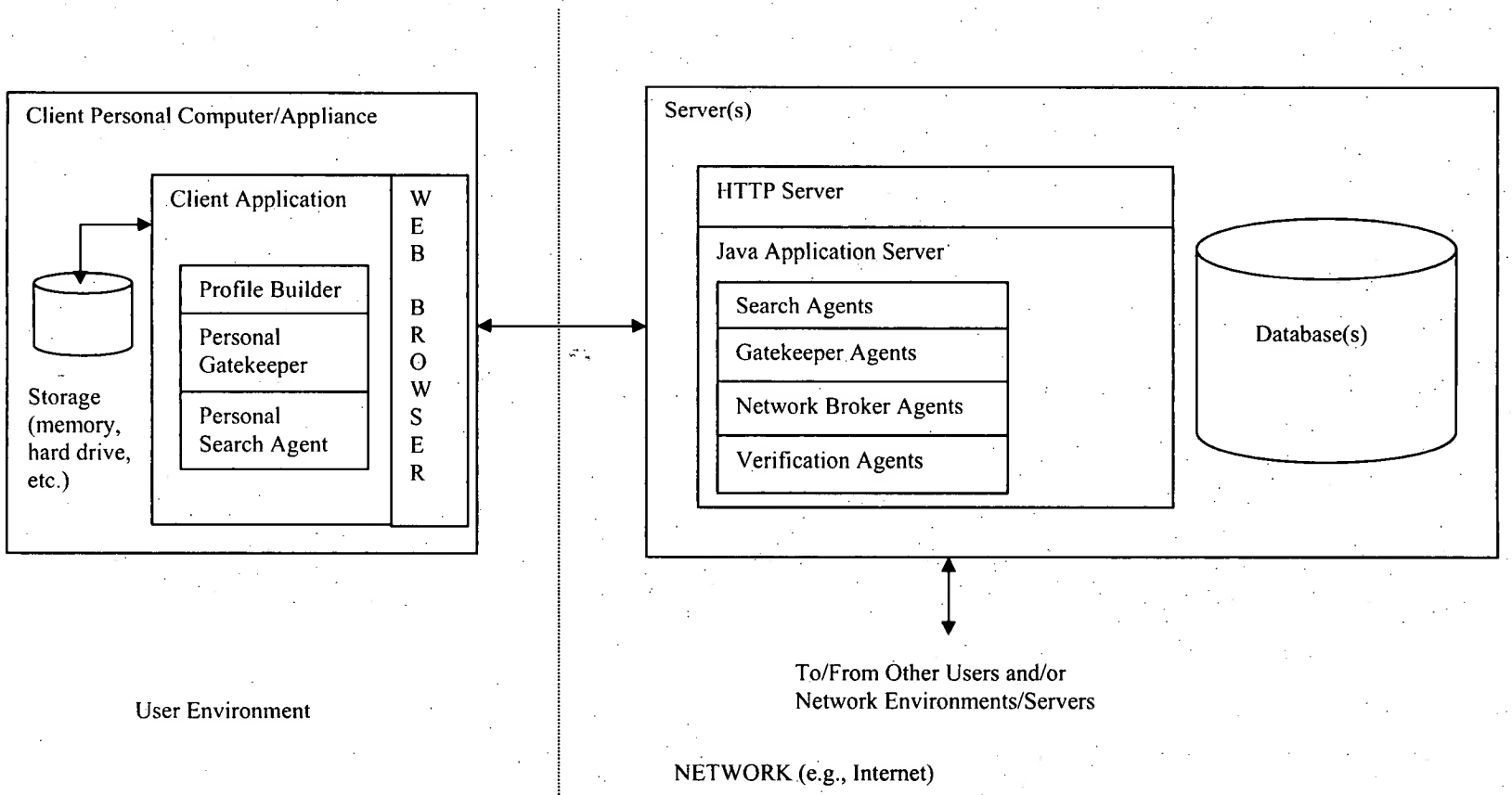

Client Personal Computer/Appliance
Client Application
Profile Builder
Personal Gatekeeper
Personal Search Agent
WEB BROWSER
Storage (memory, hard drive, etc.)
User Environment
Server(s)
HTTP Server
Java Application Server
Search Agents
Gatekeeper Agents
Network Broker Agents
Verification Agents
Database(s)
To/From Other Users and/or Network Environments/Servers
NETWORK (e.g., Internet)

Fig. 1

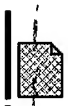
EXHIBIT C
To the declaration of James Duncan Work

Duncan Wk
duncan@netdeva.com
>

04/28/2000 01:59 PM
Please respond to
duncan

To: Tarek Fahmi/Bstz

Subject: Documents attached



- att-1.htm

- NetDevaDocs.zip

EXHIBIT D
To the declaration of James Duncan Work

Tarek Fahmi
05/09/2000 04:04 PM

To: duncan@netdeva.com
cc:
Subject: provisional application

Encl. is a copy of the provisional as drafted. let me know if you'd like to see any changes.

Fig 1.doc Provisional Application.do

Regards,
Tarek N. Fahmi
Blakely, Sokoloff, Taylor & Zafman LLP
1279 Oakmead Parkway
Sunnyvale, CA 94086-4039
Tel: 408.720.8300
Fax: 408.720.9397